isEmployeeDemo

Providing and Consuming a Web Service Between Two PeopleSoft Systems.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>M_HM_EMPLOYEE_DEMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Alias</td>
<td>EmployeeDemo</td>
</tr>
<tr>
<td>Service Operation Name</td>
<td>M_HM_IS_EMPLOYEE_DEMO</td>
</tr>
<tr>
<td>Service Operation Alias</td>
<td>isEmployeeDemo</td>
</tr>
<tr>
<td>Request Message Name</td>
<td>M_HM_IS_EMPLOYEE_DEMO</td>
</tr>
<tr>
<td>Request Message Alias</td>
<td>isEmployeeDemo</td>
</tr>
<tr>
<td>Request Message Namespace</td>
<td><a href="http://mais.he.umich.edu/schemas/isEmployeeDemo.v1">http://mais.he.umich.edu/schemas/isEmployeeDemo.v1</a></td>
</tr>
<tr>
<td>Response Message Name</td>
<td>M_HM_IS_EMPLOYEE_DEMO_RESP</td>
</tr>
<tr>
<td>Response Message Alias</td>
<td>isEmployeeDemoResponse</td>
</tr>
<tr>
<td>Response Message Namespace</td>
<td><a href="http://mais.he.umich.edu/schemas/isEmployeeDemoResponse.v1">http://mais.he.umich.edu/schemas/isEmployeeDemoResponse.v1</a></td>
</tr>
<tr>
<td>Handler Details</td>
<td>M_HM_EMPLOYEE_DEMO:isEmployeeDemo:OnRequest</td>
</tr>
<tr>
<td>Request Schema URL</td>
<td><a href="http://hedev.dsc.umich.edu/PSIGW/PeopleSoftServiceListeningConnector/isEmployeeDemo.v1.xsd">http://hedev.dsc.umich.edu/PSIGW/PeopleSoftServiceListeningConnector/isEmployeeDemo.v1.xsd</a></td>
</tr>
<tr>
<td>Response Schema URL</td>
<td><a href="http://hedev.dsc.umich.edu/PSIGW/PeopleSoftServiceListeningConnector/isEmployeeDemoResponse.v1.xsd">http://hedev.dsc.umich.edu/PSIGW/PeopleSoftServiceListeningConnector/isEmployeeDemoResponse.v1.xsd</a></td>
</tr>
<tr>
<td>WSDL URL</td>
<td><a href="http://hedev.dsc.umich.edu/PSIGW/PeopleSoftServiceListeningConnector/EmployeeDemo.1.wsdl">http://hedev.dsc.umich.edu/PSIGW/PeopleSoftServiceListeningConnector/EmployeeDemo.1.wsdl</a></td>
</tr>
</tbody>
</table>

SQL to determine if an employee:

select * from ps_employees where emplid = '12345678';
OR
select * from ps_employees where emplid = (select emplid from ps_person_sa where campus_id = 'UNIQNAME');

Sample Request XML:

• Input is UMID or Uniqname.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<isEmployeeDemo>
    <UMID>12345678</UMID>
    <Uniqname/></Uniqname>
</isEmployeeDemo>
```

Request Schema:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://mais.he.umich.edu/schemas/isEmployeeDemo.v1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://mais.he.umich.edu/schemas/isEmployeeDemo.v1"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <xs:element name="isEmployeeDemo">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="UMID" type="xs:string" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```
Sample Response XML:

- Return yes/no indicator for employee status.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<isEmployeeDemoResponse>
  <isEmployee>Yes</isEmployee>
</isEmployeeDemoResponse>
```

Response Schema:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://mais.he.umich.edu/schemas/isEmployeeDemoResponse.v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://mais.he.umich.edu/schemas/isEmployeeDemoResponse.v1"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="isEmployeeDemoResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="isEmployee">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Yes"/>
              <xs:enumeration value="No"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Build steps for making service available:

1. Create XML schemas (use editor of your choice).
2. Create request and response messages.
3. Create application class and PeopleCode.
4. Create service.
5. Create service operation.
6. Link request and response messages.
7. Generate any-to-local routing.
8. Link handler code.
9. Grant PeopleSoft Service Operation security.
11. Test.
Steps to consume a web service:

1. Consume Web Service Wizard. This will create a service, service operation, messages and a routing all at once. Example:
   - Created/Updated Operation: ISEMPLOYEEDEMO.
   - Created Request Message: M656317668.
   - Generated schema for Message: M656317668
   - Created Response Message: M666168244.
   - Generated schema for Message: M666168244
   - Created Routing: ~IMPORTED~19687 for Operation: ISEMPLOYEEDEMO
   - Created Operation Version: V1

2. Grant PeopleSoft Service Operation security.
3. Test using Service Operation Tester. You will need to provide sample request XML.
4. Add code to your page, component, app engine or whatever needs to call the service operation.